

作业提交服务器: `ftp://192.168.134.123`

用户名: `uploader`

密码: `sdt7%5252@3`

### ! 注意事项

- 每题需要提交一个cpp文件，确保cpp文件可以正确编译。cpp文件用题目编号命名。
- 每题需要提交至少一张运行结果截图。将所有题目的截图放在一个pdf文件中。
- 将所有cpp文件和一个pdf文件打包成zip文件，用自己的学号命名，并上传至ftp指定文件夹。

举例说明：比如学号是1001，本次有3个题目，那么最终提交1个`1001.zip`文件，其中包含3个cpp文件（分别是`1.cpp`, `2.cpp`, `3.cpp`）和1个pdf文件，其中pdf文件中包含至少3个运行结果的截图。第1次作业上传至hm1文件夹。

## 第1次作业

提交时间: 2024年4月7日22:00

### 第一题问题描述

西西艾弗岛上共有 $n$ 个仓库，依次编号为 $1, \dots, n$ 。每个仓库均有一个 $m$ 维向量的位置编码，用来表示仓库间的物流运转关系。具体来说，每个仓库 $i$ 均可能有一个上级仓库 $j$ ，满足：仓库 $j$ 位置编码的每一维均大于仓库 $i$ 位置编码的对应元素。比如编码为 $(1, 1, 1)$ 的仓库可以成为 $(0, 0, 0)$ 的上级，但不能成为 $(0, 1, 0)$ 的上级。如果有多个仓库均满足该要求，则选取其中编号最小的仓库作为仓库 $i$ 的上级仓库；如果没有仓库满足条件，则说明仓库 $i$ 是一个物流中心，没有上级仓库。

现给定 $n$ 个仓库的位置编码，试计算每个仓库的上级仓库编号。

### 输入格式

从标准输入读入数据。输入共 $n + 1$ 行。输入的第一行包含两个正整数 $n$ 和 $m$ ，分别表示仓库个数和位置编码的维数。接下来 $n$ 行依次输入 $n$ 个仓库的位置编码。其中第 $i$ 行 ( $1 \leq i \leq n$ ) 包含 $m$ 个整数，表示仓库 $i$ 的位置编码。

### 输出格式

输出到标准输出。输出共 $n$ 行。第 $i$ 行 ( $1 \leq i \leq n$ ) 输出一个整数，表示仓库 $i$ 的上级仓库编号；如果仓库 $i$ 没有上级，则第 $i$ 行输出0。

### 样例输入

1	4	2
2	0	0
3	-1	-1
4	1	2
5	0	-1

### 样例输出

1	3
2	1
3	0
4	3

### 样例解释

对于仓库  $2 : (-1, -1)$  来说，仓库  $1 : (0, 0)$  和仓库  $3 : (1, 2)$  均满足上级仓库的编码要求，因此选择编号较小的仓库  $1$  作为其上级。

### 子任务

50% 的测试数据满足  $m = 2$ ；

全部的测试数据满足  $0 < m \leq 10$ 、 $0 < n \leq 10$ ，且位置编码中的所有元素均为绝对值不大于  $10^6$  的整数。

### 题目背景

质数（又称“素数”）是指在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数的自然数。

### 第二题问题描述

小 P 同学在学习了素数的概念后得知，任意的正整数  $n$  都可以唯一地表示为若干素因子相乘的形式。如果正整数  $n$  有  $m$  个不同的素数因子  $p_1, p_2, \dots, p_m$ ，则可以表示为： $n = p_1^{t_1} \times p_2^{t_2} \times \dots \times p_m^{t_m}$ 。

小 P 认为，每个素因子对应的指数  $t_i$  反映了该素因子对于  $n$  的重要程度。现设定一个阈值  $k$ ，如果某个素因子  $p_i$  对应的指数  $t_i$  小于  $k$ ，则认为该素因子不重要，可以将  $p_i^{t_i}$  项从  $n$  中除去；反之则将  $p_i^{t_i}$  项保留。最终剩余项的乘积就是  $n$  简化后的值，如果没有剩余项则认为简化后的值等于 1。

试编写程序处理  $q$  个查询：每个查询包含两个正整数  $n$  和  $k$ ，要求计算按上述方法将  $n$  简化后的值。

### 输入格式

从标准输入读入数据。输入共  $q + 1$  行。输入第一行包含一个正整数  $q$ ，表示查询的个数。接下来  $q$  行每行包含两个正整数  $n$  和  $k$ ，表示一个查询。

### 输出格式

输出到标准输出。输出共  $q$  行。每行输出一个正整数，表示对应查询的结果。

### 样例输入

1	3
2	2155895064 3
3	2 2
4	10000000000 10

### 样例输出

1	2238728
2	1
3	10000000000

### 样例解释

查询一： $n = 2^3 \times 3^2 \times 23^4 \times 107$ , 其中素因子3指数为2, 107指数为1。将这两项从n中除去后, 剩余项的乘积为  $2^3 \times 23^4 = 2238728$ 。

查询二：所有项均被除去，输出1。

查询三：所有项均保留，将n原样输出。

### 子任务

40% 的测试数据满足： $n \leq 1000$ ;

80% 的测试数据满足： $n \leq 10^5$ ;

全部的测试数据满足： $1 < n \leq 10^{10}$  且  $1 < k, q \leq 10$ 。

## 第三题问题描述

对于平面直角坐标系上的坐标 $(x, y)$ , 小P 定义了一个包含 $n$ 个操作的序列 $T = (t_1, t_2, \dots, t_n)$ 。其中每个操作 $t_i(1 \leq i \leq n)$ 包含两个参数 $dx_i$ 和 $dy_i$ , 表示将坐标 $(x, y)$ 平移至 $(x + dx_i, y + dy_i)$ 处。现给定 $m$ 个初始坐标, 试计算对每个坐标 $(x_j, y_j)(1 \leq j \leq m)$ 依次进行  $T$ 中  $n$ 个操作后的最终坐标。

### 输入格式

从标准输入读入数据。输入共 $n + m + 1$ 行。输入的第一行包含空格分隔的两个正整数 $n$ 和 $m$ , 分别表示操作和初始坐标个数。接下来 $n$ 行依次输入 $n$ 个操作, 其中第 $i(1 \leq i \leq n)$ 行包含空格分隔的两个整数 $dx_i$ 和 $dy_i$ 。接下来 $m$ 行依次输入 $m$ 个坐标, 其中 $j(1 \leq j \leq m)$ 行包含空格分隔的两个整数 $(x_j, y_j)$ 。

### 输出格式

输出到标准输出中。输出共 $m$ 行, 其中第 $j(1 \leq j \leq m)$ 行包含空格分隔的两个整数, 表示初始坐标 $(x_j, y_j)$ 经过 $n$ 个操作后的位置。

### 样例输入

1	3 2
2	10 10
3	0 0
4	10 -20
5	1 -1
6	0 0

### 样例输出

1	21 -11
2	20 -10

## 样例说明

第一个坐标 $(1, -1)$ 经过三次操作后变为 $(21, -11)$ ；第二个坐标 $(0, 0)$ 经过三次操作后变为 $(20, -10)$ 。

## 评测用例规模与约定

全部的测试数据满足： $n, m \leq 100$ ，所有输入数据 $x, y, dx, dy$ 均为整数且绝对值不超过100000。

## 第四题问题描述

对于平面直角坐标系上的坐标 $(x, y)$ ，小P定义了如下两种操作：

1. 拉伸 $k$ 倍：横坐标 $x$ 变为 $kx$ ，纵坐标 $y$ 变为 $ky$ ；
2. 旋转 $\theta$ ：将坐标 $(x, y)$ 绕坐标原点 $(0, 0)$ 逆时针旋转 $\theta$ 弧度 $(0 \leq \theta < 2\pi)$ 。易知旋转后的横坐标为 $x \cos \theta - y \sin \theta$ ，纵坐标为 $x \sin \theta + y \cos \theta$ 。

设定好了包含 $n$ 个操作的序列 $(t_1, t_2, \dots, t_n)$ 后，小P又定义了如下查询：

- $i \ j \ x \ y$ ：坐标 $(x, y)$ 经过操作 $(t_i, \dots, t_j)(1 \leq i \leq j \leq n)$ 后的新坐标。

对于给定的操作序列，试计算 $m$ 个查询的结果。

## 输入格式

从标准输入读入数据。输入共 $n + m + 1$ 行。输入的第一行包含空格分隔的两个正整数 $n$ 和 $m$ ，分别表示操作和查询个数。接下来 $n$ 行依次输入 $n$ 个操作，每行包含空格分隔的一个整数（操作类型）和一个实数（ $k$ 或 $\theta$ ），形如 $1 k$ （表示拉伸 $k$ 倍）或 $2 \theta$ （表示旋转 $\theta$ ）。接下来 $m$ 行依次输入 $m$ 个查询，每行包含空格分隔的四个整数 $i, j, x, y$ ，含义如前文所述。

## 输出格式

输出到标准输出中。输出共 $m$ 行，每行包含空格分隔的两个实数，表示对应查询的结果。

## 样例输入

```
1 10 5
2 0.59
3 2 4.956
4 1 0.997
5 1 1.364
6 1 1.242
7 1 0.82
8 2 2.824
9 1 0.716
10 2 0.178
11 2 4.094
12 1 6 -953188 -946637
13 1 9 969538 848081
14 4 7 -114758 522223
15 1 9 -535079 601597
16 8 8 159430 -511187
```

## 样例输出

```
1 -1858706.758 -83259.993
2 -1261428.46 201113.678
3 -75099.123 -738950.159
4 -119179.897 -789457.532
5 114151.88 -366009.892
```

## 样例说明

第五个查询仅对输入坐标使用了操作八：拉伸 0.716 倍。

横坐标： $159430 \times 0.716 = 114151.88$

纵坐标： $-511187 \times 0.716 = -366009.892$

由于具体计算方式不同，程序输出结果可能与真实值有微小差异，样例输出仅保留了三位小数。

## 评测用例规模与约定

80% 的测试数据满足： $n, m \leq 1000$ ；

全部的测试数据满足：

- $n, m \leq 100000$ ；
- 输入的坐标均为整数且绝对值不超过 1000000；
- 单个拉伸操作的系数  $k \in [0.5, 2]$ ；
- 任意操作区间  $t_i, \dots, t_j (1 \leq i \leq j \leq n)$  内拉伸系数  $k$  的乘积在  $[0.001, 1000]$  范围内。

## 评分方式

如果你输出的浮点数与参考结果相比，满足绝对误差不大于 0.1，则该测试点满分，否则不得分。